

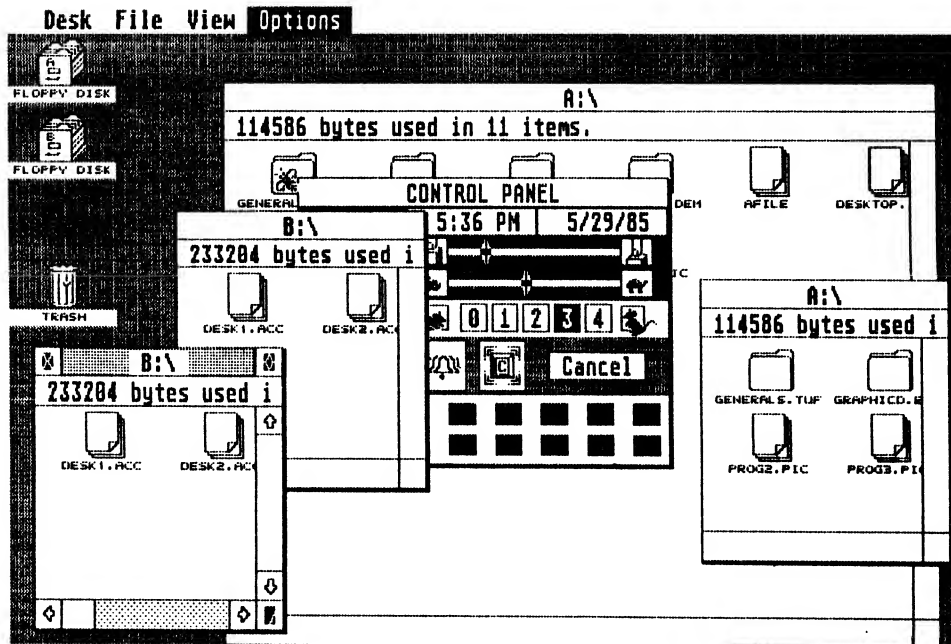
# ATARI COMPUTER ENTHUSIASTS

3662 Vine Maple Dr. Eugene OR 97405

SEPTEMBER, 1985

Mike Dunn, Jim Bumpas & Larry Gold, Editors

## GUESS?



## DOCKERS AND LUTZ ARE HERE AGAIN

# News and Reviews

by Mike Dunn, Co-Editor

Many of our members bought the new Atari 520ST computers and are very happy with them. Some did not work after the trip overseas, but all could be fixed by re-seating the chips. Atari is to be congratulated for helping user groups to fix their computers and giving advice over the phone on how to do so. A letter just received from Atari states that BASIC is almost ready to ship as well as the upgrade ROM; both will be sent free of charge to those who bought their ST through the club. Jim Bumpas will be in charge of the ST info since he bought one and I didn't. Software for the ST is being announced at a rapid rate; the newest is by **Batteries Included** (17875 Skypark North, Suite P, Irvine, CA 92714.) Called **The Isgur Portfolio System** it is a very elaborate stock market program, including tax implications, what-if abilities, Telecommunications, etc. It costs \$250 and is for the IBM-PC and the 520ST.

Two interesting new devices are announced this month. One is **Computereyes** (Digital Vision, 14 Oak St #2, Needham, MA 02192.), a \$130 interface to attach a video camera or TV to your Atari to obtain digitized pictures. Larry Gold will report more about this. The other is \$3500 fingerprint biometric verification and identification system that reads your fingerprints and verifies it before you can use your computer. Called the **RidgeReader** (Command Productions, 62 Bowman, Rye Brook, NY 10573) it is just the thing to keep the kids from playing games on your Atari when they should be studying.

TAB Books (POB 40, Blue Ridge Summit, PA 17214) has just released their new 96 page catalog of technical books. If you're are not familiar with TAB, they put out technical books on just about any subject you can think of, from computers to the story of the Lotus car; from books on the Atari to how to repair clocks. An example is a book they recently sent to me to review — **Computer Technicians Handbook** by Art Margolis (\$17, 500 pages). It begins by explaining how computers work, truth tables, binary arithmetic, how the various chips including CPU's, interface chips, logic devices, memories, etc. work. It also explains machine language and gives hints on how to trouble shoot and repair computers. This is not a step-by-step how to do it guide; the book tries to explain how everything works so you can fix the computer by understanding what is happening.

MPP (Microbits) of Albany, OR has closed down and reopened with the same principles as **Supra Corp** 1133 Commercial Way, Albany, OR 97321, (503)967-9075. They will continue to market MPP products. Kirt Stockwell is no longer with them, but has formed a new company **SofMarc** (4325 Sean, Eugene, OR 97402 / 503-689-3565, BBS 689-2348) Their first product is our new Bulletin Board System **ESP Mindlink** - see Larry Gold's article.

**Batteries Included**, the Canadian software company, has dropped plans to produce the 80 column cartridge for **PaperClip** because of difficulty of obtaining the proper chips in a timely manner. Incidentally, I've also been having problems with dropped characters with PaperClip — anyone else having the problem?

\*\*\*\*\*

We have many new disks to add to the library this month thanks to all the great programs our members keep sending in and also from other clubs we trade with. The quality of the public domain programs is becoming quite remarkable, and some of the disks are of commercial quality. The single subject disks all have built in documentation and/or help facilities.

**Best of ACE -15** (\$10). The latest programs from the newsletter, it includes SkyPlot, the long astronomy program from our pres Dick Barkley which locates and pictures objects in the night sky, run time versions of Stan Ocker's Action! program Double Bounce as extensively modified by Rick Groszkiewitz (who incidentally sent us four disk sides full of Action! programs and articles for the next few issues), and other programs from the last couple of issues of ACE.

**ACTION! Disk -4** (\$10). Has both source code and run-time versions of the recent ACTION! programs in ACE including this month's ROWS by Stan Ockers (but not in run-time since it wouldn't work), XECOPY and FID by Dale Lutz, and several variations of Stan's BOUNCE program by Rick Groszkiewitz. Rick finds BOUNCE to be his favorite game, so he has modified it for one player, one joystick, etc. There are also some of the utilities found in the last few issues.

**ACE Business Disk -2: Data Manager** by John Logan (\$10) For you **BASIC XL** owners, here is a very nicely done data base system which even has a built in tutorial. We could supply it with the BASIC XL runtime for those who do not own the O.S.S. cartridge; but I don't know how much memory that might take up and interfere with the program. Worth getting just to see the various programming techniques used if you have BASIC XL.

**ACE Business Disk -3: General Ledger** (\$10) by Neville Williams. This General Ledger program is designed for a small business with 99 accounts. Written by one of our Australian members, it requires you to use the European style of dating (DDMMYY). It includes an on disk manual which is very good; please see review in this issue of the disk. From other Atari Clubs:

**Wizard** \$15 a public domain Bulletin Board System.

**\*\*from Computer Squad (c/o Suzi Subeck, 2625 Corinth Road, Olympia Fields, IL 60461)** comes two of the most outstanding public domain programs I've ever seen:

**Computer Squad 3-D CAD System** (\$10). Written by Steve Ulstad, a professional arcade game designer, this machine language program allows you to draw and save screens, offset it, reflect the design, change scale on one or more axes, perform translations, change viewing angle, depth and aspect ratios, rotate, and print the images. Comes with extensive on disk documentation.

**Computer Squad CAD/CAM Facility Layout** (\$15 2 disk sides). by Jack Gilchrist and Phil Heavens. The authors are used to working on large scale graphic systems, this program is more geared to floor plans. It also has 3-D capabilities. You can also change the scaling, viewing windows, viewing angle, etc. It includes extensive on-line help facilities, examples and a large manual you can print out. It also includes a text-labeling feature. It is written in BASIC and takes up two disk sides.

## ERACE UPDATE

### DISK NO. 11

GERMAN, PORTUGUESE, FRENCH — These three language programs are similar to the Spanish and Italian on the ERACE Disk #9 but in a multiple choice format.

LEXICON — provides 200 words, their synonyms and antonyms to develop word skills.

LETTNUMB — A letter and number identification for the preschooler.

TRANSITN — TRANSITION: A Game of Logic. Only two ways to solve the puzzle. Very challenging.

TICTOC — What time is it? Learned to tell the time the old fashioned way.

HELP — Documentation for programs on this disk.

— N. F. Young  
ERACE Sig Editor

## TEASERS BY TOBBS

Teasers by Tobbs (\$55, Sunburst communications, Recommended grade levels: 4 to adult) contains two challenging programs — Addition and Multiplication. There are six levels to each program. Level 1-3 provide basic mathematic drills to reinforce mental arithmetic skills. In Levels 4-6, the puzzle is in reverse causing the student to work backward by subtracting/dividing.

The student is learning relationships between mathematical operations. Every level increases in difficulty.

Tobbs is a lovable character who appears with the drill/puzzle to guide the student in finding the missing number. There are visual responses whether an answer is correct or not. The instructions are clear and precise and included on the disk. One or two persons can play. There is a 30-page manual in a bright orange notebook with suggestions for both parent and teacher.

Joe, our 3rd grader enjoys this disk. It can be challenging yet hits the basics. The boy down the street who hates math is drawn to the higher levels. He likes to use his logic to solve the puzzles.

# XE COPY

The first thing I did after I familiarized myself somewhat with my 130XE was to write a sector copier which uses all 128K available to copy a single density disk in a single pass. This allows multiple copies to be made easily because the whole disk is read into memory at a time. Double density disks can be copied in two passes. Even though it used to be a tight squeeze to be able to copy a single density disk in two passes with the old 800, I had plenty of room left over in the 130XE to do it in a single pass.

## Using the Program:

The program is extremely easy to use. Those of you with an ANALOG speech synthesizer are in for a treat — if you hook it up before running the program, you will be audibly reminded in nearly clear English to insert the source disk, or whatever. The only keys used are the Start and Select buttons. The program is completely self prompting along the way. One point to bear in mind is when you are copying a double density disk, make sure not to forget that first you have to write out the first half of the disk on all your destination disks, then read in the other half, and write the second half out to all of the destination disks.

All your destination disks must be preformatted in the correct density before you begin copying. If an error occurs when copying, you will be given a chance to retry that sector.

As a disk is read in or written out, a bar graph of letters (R for read, W for write) goes across the screen to show you how the program is progressing. For those of us with digital track readouts, however, this is not such a big deal.

## The Program Itself

As some of you observant ones may have already noticed, XECOPY is written in beautiful structured ACTION!. ACTION! is a logical choice for this program, since it allows me easy access to operating system routines for reading a sector while at the same time allowing me to code in a high level language.

A few of the procedures may be of interest to the programmers out there. Procedure SystemCheck() could be used in any programs made only for the 130XE. This procedure actually does two checks — first it verifies the bank switching ability of the computer by storing a number in the main memory area, attempting a bank switch, and storing another number to the same memory address. The bank is then flipped back, and a check made to see if the original stored number is still intact. If so, this test is passed. The second test checks to see if a cartridge is present or not. Note that my procedure BankChange() automatically flips out the built in BASIC when it is called, so the BASIC is gone when this test is tried. All the program does is store a series of values in a location where a cartridge could be, and then it checks to see that the same value is still there. If a ROM were present, the value stuffed in would not necessarily be there when the location is read. If this happens, then this test is failed. If either test is failed, the machine locks into an infinite loop (but since location 580 is alerted, reset will cause the machine to reboot).

Those with a speech synthesizer will want to look at procedures InitSpeak() and Say(). The Say() procedure takes a byte array of allophones (whose end is marked by a 255) and dumps them out to the speech synthesizer, while the InitSpeak() procedure simply prepares the joystick ports for output.

Another procedure of note is the procedure CheckSpeak(). It checks whether or not an ANALOG speech synthesizer is hooked up so the program will talk if possible. All it does is send a delay to the synthesizer, and waits to see how long it takes to respond. If it takes too long (more than 3 jiffies), then Say() procedure is replaced by a simple delay procedure.

The HandleSector() procedure is used to set up the call to the resident disk handler. I must give credit to Roger Edwards who presented the technical information for this in the Edmonton Atari User's group newsletter. Note that when a double density sector is to be written or read, a different system jump must be made than when a single density sector is handled. Also note that this same routine both reads and writes sectors depending on the value of the BYTE direction.

## Program Notes

The program simply goes from sector 1 to sector 720 and reads the disk into memory. Bank switching is done whenever the current bank gets filled up. Notice that the procedure ReadAndWrite() looks after the buffer allocation. It both reads in and writes out the sectors by calling HandleSector(). You may notice that the first three sectors are handled specially by ReadAndWrite if the disk was double density. It turns out that on all disks, sectors one through three are in single density. These are the boot sectors which are read first when you boot up your computer.

If you wish to compile this program yourself with your ACTION! cartridge, you will have to have the Runtime package. Since the program will not run if a cartridge is installed, it is impossible to run it with the ACTION! cartridge in. Therefore, those of you without SYS.ACT will have to order the disk with XECOPY.COM on it in order to use the program. If you have SYS.ACT, make sure you compile XECOPY from off of the disk so that it resides low enough in memory in its runtime form (i.e. make sure that the editor is cleared before you compile).

## Conclusion

This program is not the ultimate in a copier. There is still some more memory room left to add all kinds of bells and whistles. However, I wanted to make a simple, easy to use sector copier to use all of the 130XEs memory for a buffer so I could make multiple copies of disks (especially double density) quickly and easily. XECOPY fits the bill nicely, and will be a very useful addition to any 130XE owner's library.

— Dale Lutz

MEETING

SOUTH EUGENE  
HIGH

WEDNESDAY  
SEPTEMBER 11TH

7:30PM

## GENERAL LEDGER ACCOUNTING SYSTEM

The General Ledger Accounting System (by Neville R. Williams of Queensland, Australia and available from ACE for \$10, including a Translator) is a well written and well documented general ledger accounting program. The author has included an excellent manual and a brief tutorial on disk. The program requires an 8-bit Atari with at least 48k, a Translator with XL/XE machines, one disk drive and a printer capable of coding tab stops.

The system limitations include: 99 accounts numbered from 1 to 99. A total of 1,000 transactions can be entered before a balance forward is required and each entry is limited to ten dissections. The program presupposes that the user has a basic understanding of the double-entry system of accounting.

As I worked with this program I was more and more impressed with the thoroughness of the programmer's work. Try as I might, I could not find any place which did not work or any area which was not well documented. However, I do want to caution those who may use the program that when the date is called for, the program requires the date be given in the following format: DDMMYY.

I want to see more accounting applications from Neville Williams as he clearly understands what is needed in an accounting program.

— Kathy Henderson

# BUMPAS REVIEWS

**GEMSTONE WARRIOR** (SSI, \$35) is an "action, strategy and adventure in the netherworld." Demons have stolen the Gemstone and separated it into 5 pieces which you must find and reassemble. The keyboard is required for many functions, but the joystick is an option for movement, firing and searching. The game may be saved, paused, or played at various skill levels.

To make your quest difficult (if not impossible!) are diseased and vicious monsters, Thieving Flyers, Skeletons, Gas Plants, Giant Amoebas, Ghosts, Summoning Eyes and the Demons themselves.

As the Brave Warrior who has taken up this challenge you will depend upon your primary weapons (arrows and magic bolts). But you will also find up to 15 different types of magic items (some of which will be inside coffins and chests) and treasure. Several of the magic items have undocumented properties — you must discover what they do (they may differ from game to game). Even the items for which you know what they do, experience will be needed to discover how to use them appropriately.

There are various doors, archways and gates into the various rooms of the underground caverns. Some places will appear to be dead ends with no way out as you return to the same rooms again and again. But perseverance, cleverness, agility and careful observation techniques will carry you through to success. This is not a game to master in one sitting!

Synapse & Broderbund have released the first "electronic novel", **MINDWHEEL** (\$?). Nothing less is at stake than the survival of civilization. Four unprotected disk sides of files are provided together with a hard-bound novelette of nearly 100 pages. You must provide a "Bookmark" disk on which to save a game in progress. If you have a printer, you can also create a printed record of your progress in the game.

Two disk drives are required, and the program also supports 52k add-on RAM boards, Mosaic and Axlon Ramdisks. A Happy-modified disk drive is also specifically mentioned for those who might want faster loading and responses.

Basically a text adventure, the program comes closer to a "natural language" game than earlier text adventures do. I believe it recognizes close to 1200 words. You need to read the book to learn as much as possible about the four main characters in the game. You must "travel" through the minds of these four: Bobby Clemon, an assassinated rock star; The Generalissimo, a dictator and war criminal; The Poet, composer of epic literary achievements; and Dr. Eva Fein, scientist, humanist and distinguished musician.

The object of your quest is to find The Cave Master and the Wheel of Wisdom which he guards.

I'm having more fun with **COLONIAL CONQUEST** (SSI, \$40) than I've had with any computer game in a long time. It's a very social game — best played with 6 players. But the computer will play any of the 6 positions not used by one of your friends. Each computer position may be separately set at one of 10 levels of difficulty. Players control one or more of 6 colonial powers: England, Germany, France, USA, Japan and Russia in one of three scenarios: Standard (no one starts with any colonies), and two historical scenarios (1880 and 1914).

The map scrolls across 4 screens of very good graphics containing 131 land areas plus the oceans. Each player may build and move armies and fleets to attack or garrison possessions. Land zones may be fortified. You may pay for espionage. The information you get may not be entirely accurate, but will give you some indication of the economic value of a zone and the number of its defenders. You may also subvert the armies in zones not controlled by other players in order to control the zones outright, or to make conquest by force easier. Economic aid may be given to any other player or to a minor country.

The real strength of this game is the opportunity and necessity for diplomacy to help you win. Especially Russia and Japan need skills in this area to do well. If you just fight all the time you'll find your forces depleted and so cannot be much of a threat to anyone. On the other hand, if you're too threatening, anyone can be wiped out by a combination of other players.

Each player spends an annual budget of income from the possession of land zones and gifts of economic aid from other players. Control of all the zones in a larger region (i.e., South America or the Far East) may produce big bonuses of income.

As a grand strategic multi-player game of diplomacy, treachery and "force majeure" this game is unexcelled.

I've just begun using the new **Prentice P-212ST** modem (after selling my 300 baud Hayes). This 1200 baud modem sells for less than \$300 and advertizes "Hayes compatibility".

If you are used to all the lights on a Hayes, you might miss them. The Prentice has only a power light and a carrier detect light. You can set the internal speaker to be on at all times, so you should be able to hear some data noises. There are no dip switches to set: all modem functions are software controlled and can be set by commands from terminal mode in your communications software.

I've been using Amodem quite a lot lately, but so far I've been unable to use the Autodial Directory which was a nice feature of the program when I used it with the Hayes. I've set the software switch (ATS18=0) to emulate the Hayes. But when the autodial selection is made the screen tells me it's dialing, and then that it's made a connection. But no dialing occurs, and no connection is made. When I dial manually on the keyboard in terminal mode I have no trouble making connections.

I never realized how much I would enjoy seeing a whole screenful of information flash on the screen at once. But after watching the cursor paint across the screen at 300 baud, 1200 baud seems extremely fast. I can't even imagine 2400!

— Jim Bumpas, Co-Editor

## VP's RAMBLINGS

The summer is about to end, school is ready to start and Atari Inc. is starting to go full blast with the ST, XE and anything else that they have up their sleeve. The stores are beginning to have and sell new programs and the future looks better for Atari than it has in a long time.

One company which hasn't been sitting back and waiting is Broderbund. They have a new graphics disk for the Print Shop program. This is the second graphics disk for that program and it is just as good as the one before and makes the choice of graphics that much more for the user to choose from when he uses the program. The newsletter has from time to time used these graphics when it was appropriate to the issue. Broderbund should be congratulated for continuing to make a fine program better and more useful.

I wish more companies would update their programs or enhance them with more features as has Broderbund. They have done the same with templates that they have brought out for Synapse's SynCalc. I will talk more about this in another article.

We are Beta testing a BBS program for a new company called Sofmark. The board is almost to its last stage of development so please bear with us for any problems you may find. We want to have your comments on the board and also let us know if there are any features you would like to see incorporated into the board which are not there. When it does come out it will be called BBS Mindlink.

The BBS is easy to use as all you have to do when you call up is give it your name and after a short pause it will ask for your phone no., then four character password. Please be careful as the board is written in machine language and is very fast in its response time. Once you have signed on and have gone through the board be sure that you answer the prompt about saving your password or you will have to sign on again until you have saved your password. I can't upgrade your level to have access to the whole board unless you log your password. I hope you like it and will let us know what you think about it.

— Larry Gold

## RAMDISK

The new Atari 130XE has an extra 64K of bank memory which can be utilized as a RAM disk which will run 20 times faster than a normal floppy disk. The program TMDRIVE.BAS can be used in conjunction with the Atari Translator to give you a single density disk emulator with 523 free sectors. It should work with any DOS system which uses the Atari file management system (most do).

To run the program, first boot the translator disk, then boot whatever DOS system you normally use. Now run the program from BASIC which will automatically set up the drive emulator (be sure you have saved the program first, as running it will destroy it). Type NEW before continuing. You can now address this RAM disk as D4:.

Do not try to format the RAM disk as it will indicate more free sectors than there actually are. If you need to reformat it, then just run the program again. If you are using Atari DOS 2.0 then you can put DUPSYS and MEM.SAV onto D4: First copy the files you want to D4: (DUPSYS, MEM.SAV, etc.). Then from Basic type:

```
IF PEEK(6192)=49 THEN POKE 6192,52.  
IF PEEK(5948)=49 THEN POKE 5948,52.
```

— Ralph Walden

# XE DISK COPIER

# BY DALE LUTZ

```

; XE disk copier
; version as of 7:00 June 18
; by Dale Lutz
;System Equates
Include"D8:sys.act"
DEFINE Read="$52",Write="$50",double="
2",
    single="1",start="6",select="5"
    codetop="14000",newtop="48153"
BYTE DDEVIC=$300,
    DUNIT=$301,
    DCOMMD=$302,
    DSTATS=$303,
    CONSOL=53279,
    pactl=54018,pdat=54016,
    mode,doublecode,direction,
    flasher=710,bank,realmode,grapher
CARD DBUFFER=$304,
    SECTOR=$30A,
    NUMBYTES=$308,buftop,secbegin,
    secend
BYTE ARRAY dest = [ 21 7 55 17 0 12
    12 0 11 20 37 15 11 0 255 ],
    source = [ 55 55 50 55 55 0 255 ],
    disk = [ 21 12 55 41 2 2 2 2 9 39 7
    7 55 55 2 55 13 59 17 0 255 ],
    insert = [ 12 12 11 2 55 55 52 17 0 25
    5 ],
    name = [ 4 2 2 20 6 2 2 33 20 62 2 2
    45 31 17 55 0 255 ],
    intro = [ 55 55 12 12 44 61 62 2 9
    26 26 55 55 2 33 12 12 55 41 2 8 24
    9 19 51 2 2 2 0 255 ]
PROC DSKINV=$E453()
RETURN
PROC SIOV=$E459()
RETURN
PROC GoBack() ;stub used to return to
RETURN ;handlesector after erro
r
PROC Restart() ;stub to rerun the prog
RAM
RETURN ;after error
PROC SetScreen()
Graphics(10)
PutDE(6)
PrintDE(6," now")
IF direction=read THEN
    PrintDE(6," READING")
ELSE
    PrintDE(6," WRITING")
FI
PutDE(6)
PutDE(6)
RETURN
PROC MyError()
BYTE a
Graphics(10)
PutDE(6)
PrintDE(6," error encountered")
PutDE(6)
PrintDE(6," START")
PrintDE(6," RETRY")
PutDE(6)
PrintDE(6," SELECT")
PrintDE(6," exit")
PutDE(6)
WHILE consol()start AND consol() sel
ect DO
    flasher==+1
    OD
    IF consol=start THEN
        SetScreen()
        FOR a=1 TO (sector/grapher) DO
            IF direction=read THEN
                PutD(6,'R')
            ELSE
                PutD(6,'W')
            FI
            OD
            GoBack()
        ELSE
            Restart()
        FI
    RETURN
PROC HandleSector()
DDEVIC=$31 ;disk device
DUNIT=1 ;drive #1
DCOMMD=direction ;set for read or w
ite
IF mode=double THEN
    NUMBYTES=$100
    DSTATS=doublecode ;either $40 for
;write, or $80 for read
    SIOV() ;go do it
ELSE
    DSKINV() ;go do it
FI
IF DSTATS()1 THEN
    MyError()
FI
RETURN
PROC InitSpeak()
BYTE p
p=pactl
pactl=p-4
pdat=127
pactl=p
RETURN
PROC Say(BYTE ARRAY word)
BYTE a
a=0
WHILE word(a)<255 DO
    WHILE pdat>128 DO OD
    pdat=word(a)+64
    pdat=word(a)
    a==+1
    OD
RETURN
PROC Delay(BYTE ARRAY word)
CARD a
FOR a=1 TO 10000 DO OD
RETURN
PROC CheckSpeak()
BYTE clock=20 ;this procedure
;checks that a
pdat=4+64 ;speech synth is
pdat=4 ;hooked up...
clock=0
WHILE pdat>128 DO
    IF clock>3 THEN
        Say=Delay
    RETURN

```

# XE CON'T

```

FI
OD
RETURN

PROC WaitForPress()

Graphics(18)
PutDE(6)
PrintDE(6,"    insert")
PrintDE(6,"    SOURCE")
PrintDE(6,"    DISK")
PutDE(6)
PutDE(6)
PrintDE(6,"    press START")
consol=8
IF direction=Read THEN
    Say(insert)
    Say(source)
    Say(disk)
ELSE
    Position(3,2)
    PrintD(6,"DESTINATION")
FI
WHILE consol=7 DO
    flasher==+1
OD
RETURN

7C ChangeDensity()

Position(10,11)
IF mode=single THEN
    PrintD(6,"DOUBLE")
    mode=double
    grapher=20
ELSE
    mode=single
    PrintD(6,"SINGLE")
    grapher=40
FI
realmode=mode
WHILE consol<>7 AND consol<>8 DO
    consol=8
OD
RETURN

PROC BankChange()

BYTE selector=54017,nomal

    normal=128+64+2+1 ;base on which to
                        ;add the bank sele

ct
;values
IF bank=5 THEN
    selector=255
ELSE
    selector=normal+0*16+bank*4
FI
RETURN

PROC ReadAndWrite()

;first read the thing in
direction=Read
doublecode=$40
DO
    bank=5
    BankChange()
    SetScreen()
    buftop=memtop
    DBUFFER=codetop
    FOR SECTOR=secbegin TO secend DO
        IF realmode=double THEN
            IF sector<4 THEN
                mode=single
            ELSE
                mode=double
            FI
        FI
        HandleSector()
        IF (SECTOR MOD grapher)=0 THEN
            IF direction=read THEN
                PutD(6,'R')
            ELSE
                PutD(6,'W')
            FI
        FI
        DBUFFER=DBUFFER+128*mode
        IF DBUFFER>(buftop-257) THEN
            IF bank=5 THEN
                bank=0
            ELSE
                bank==+1
            FI
            BankChange()
            DBUFFER=16384
            buftop=32767
        FI
    OD ;end of FOR
    direction=write ; now set up for ...
    doublecode=$80 ; writing it out.
    consol=8
    Graphics(18)
    PutDE(6)
    PrintDE(6,"    START")

    PrintDE(6,"    write it out")
    PutDE(6)
    PrintDE(6,"    SELECT")
    PrintDE(6,"    exit")
    Say(insert)
    Say(dest)
    Say(disk)
    PutDE(6)
    WHILE consol<>start AND consol<>s
    elect DO
        flasher==+1
        OD
        IF consol=start THEN
            WaitForPress()
        ELSE EXIT
        FI
    OD ;end of <nearly> infinite loop
    RETURN

PROC CopyDisk()

WaitForPress()
IF mode=double THEN
    secbegin=1 secend=360
    ReadAndWrite()
    direction=Read
    WaitForPress()
    secbegin=361 secend=720
    ReadAndWrite()
ELSE
    secbegin=1 secend=720
    ReadAndWrite()
FI
RETURN

PROC SystemCheck()

BYTE testloc=16385,flag,num

    flag=0
    testloc=22
    bank=2
    BankChange()
    testloc=99
    bank=5
    BankChange()
    IF testloc<>22 THEN
        PrintE("    Unable to Bank Switch")
    FI
    FOR num=1 TO 55 DO
        Poke($A004,num)
        IF Peek($A004)<>num THEN
            PrintE("    Cartridge Present..")
        FI
    OD

```

# XE CON'T

# KANGA CON'T

```

.")
  flag=1
  num=66
  FI
  OD
  IF flag=1 THEN
    PrintE("      Program aborted")

    DO OD ;infinite loop
  FI
  RETURN ;integrity test over...

PROC Copier()

  Goback=HandleSector
  Restart=Copier
  InitSpeak()
  Poke(580,3) ;set it up so that
  CheckSpeak() ;reset will reboot
  bank=5
  BankChange()
  Poke(106,192)
  SystemCheck() ;verify correct system
  ...
  Say(intro)
  Say(name)
  DO
    bank=5
    BankChange()
    Poke(106,192)
    mode=Single
    realMode=mode
    direction=Read
    Graphics(18)
    grapher=48
    consol=8
    PrintE(6,"      single pass")
    PrintE(6,"      disk copier")
    PrintE(6,"      FOR 130XE ")
    PrintE(6,"      BY      ")
    PrintE(6,"      dale lutz ")
    PutE(6)
    PrintE(6,"      START")
    PrintE(6,"      begin copying")
    PrintE(6,"      SELECT")
    PrintE(6,"      change density")
    PutE(6)
    PrintE(6,"DENSITY:  SINGLE")
    DO
      flasher==+1
      IF consol=select THEN
        ChangeDensity()
      FI
      UNTIL consol=start OD
      CopyDisk()
    OD
  RETURN

```

```

1125 FOR M=1 TO 150:SOUND 0,M,10,10:ME
XT M:SOUND 0,0,0,0
1130 RETURN
1135 REM EDIT CHARACTER SET
1140 RESTORE 1190
1145 MEMTOP=PEEK(106);GRTOP=MEMTOP-4
1150 POKE 106,GRTOP:GRAPHICS 1+16:CHRO
M=PEEK(756)*256
1155 DIM A(8),R$(1)
1160 CHRAM=GRTOP*256:POKE 756,GRTOP
1165 FOR N=0 TO 1023:POKE CHRAM+N,PEEK
(CHROM+N):NEXT N
1170 FOR K=1 TO 30:READ R$:R=ASC(R$)
1175 IF R<32 THEN R=R+64
1180 IF R<96 THEN IF R>31 THEN R=R-32
1185 FOR I=0 TO 7:READ A:A(I)=A:B=I+R*
8+256*GRTOP:POKE B,A:NEXT I:NEXT K
1190 DATA !
1195 DATA 170,254,85,253,171,251,87,24
7
1200 DATA #
1205 DATA 175,239,95,223,191,191,127,1
27
1210 DATA <
1215 DATA 10,15,21,31,42,63,85,127
1220 DATA >
1225 DATA 170,254,84,252,168,252,85,25
3
1230 DATA =
1235 DATA 170,255,85,255,170,255,85,25
5
1240 DATA $
1245 DATA 0,254,253,253,251,251,247,24
7
1250 DATA %
1255 DATA 239,239,223,223,191,191,127,
127
1260 DATA ;
1265 DATA 0,0,1,1,2,3,5,7
1270 DATA &
1275 DATA 0,0,1,1,3,3,7,7
1280 DATA '
1285 DATA 254,254,253,253,251,251,246,
247
1290 DATA )
1295 DATA 15,15,31,31,63,63,127,127
1300 DATA (
1305 DATA 237,239,218,222,180,188,104,
120
1310 DATA H
1315 DATA 76,49,130,154,89,65,140,50
1320 DATA Y
1325 DATA 147,135,36,89,154,36,17,201
1330 DATA F
1335 DATA 32,32,96,96,224,224,32,32
1340 DATA 0
1345 DATA 48,48,120,59,59,62,88,240
1350 DATA 0
1355 DATA 208,240,160,224,64,192,128,1
28
1360 DATA /
1365 DATA 0,127,127,127,127,127,127,12
7
1370 DATA .
1375 DATA 0,254,254,254,254,254,254,25
4
1380 DATA -
1385 DATA 0,255,255,255,255,255,255,25
5
1390 DATA +
1395 DATA 255,255,255,255,255,255,255,
255
1400 DATA ?
1405 DATA 0,15,15,15,15,15,15,15
1410 DATA :
1415 DATA 224,224,208,208,176,176,96,1
12
1420 DATA Z
1425 DATA 255,129,189,165,165,189,129,
255
1430 DATA X
1435 DATA 38,152,129,90,90,129,23,1
1440 DATA K
1445 DATA 238,236,248,252,238,238,0,0
1450 DATA B
1455 DATA 0,124,6,126,230,127,0,0
1460 DATA M
1465 DATA 0,252,238,238,238,239,0,0
1470 DATA G
1475 DATA 0,127,230,230,126,6,230,124
1480 DATA V
1485 DATA 153,36,66,153,153,66,36,153
1490 RETURN

```

RUTH'S LOGO

FROM LAST MONTH

```

TO PRINTOUT
PRINT [TYPE THE TITLE]
MAKE "TITLE RL
SETWRITE "P:
PR :TITLE
PR []
SETWRITE []
PRINT [TYPE YOUR NAME AND THE DATE]
MAKE "NAME RL
SETWRITE "P:
PR :NAME
PR []
SETWRITE []
PRINT [TYPE THE NAME OF THE PROCEDURE]
MAKE "PRO RL
SETWRITE "P:
PR :PRO
SETWRITE []
END

```

# ROWS AND COLUMNS

```

; ROWS AND COLUMNS
; 7-85
; 5. OCKERS

PROC RealAdd(REAL POINTER a,b,c)
    RealAssign(a,Fr0) RealAssign(b,Fr1)
    ROM_FADD() RealAssign(Fr0,c)
    RETURN
END

PROC Init() ; Fill array with _ cells
    CARD pos,size
    SetBlock(bInks,41,160) bInks(0)=40
    size=8000 SetBlock(strs,size,95) pos=s
    trs
    WHILE pos<strs+size DO
        Poke(pos,9) pos==+10
    OD
    startx=0 starty=0 maxx=0 maxy=0
    x=startx y=starty Open(1,"K:",4,0)
    modeflg=0 Put(125) errflg=0
    RETURN

PROC Blnkline() ;Inverse blank line
    Position(0,0) Print(bInks)
    Position(0,0)
    RETURN

PROC Stringpos() ;Find position
    CARD pos ;of cell in strs array
    pos=160*y+10*x
    RETURN(pos)

PROC Rowcol() ;Which screen position?
    row=2*(y-starty)+3 col=10*(x-startx)
    RETURN

PROC Prtstring() ;Put cell on screen
    CARD loc
    loc=strs+Stringpos() Rowcol()
    Position(col,row) Print(loc)
    RETURN

PROC Fillscr() ;Fill screen with cells
    BYTE savx,savy,nbr
    CARD loc
    Blnkline() savx=x savy=y
    IF modeflg=0 THEN Print("Row Entry")
    ELSE Print("Column Entry")
    FI
    savx=x savy=y
    FOR y=starty TO starty+10 DO
        IF y>49 THEN EXIT FI
        FOR x=startx TO startx+3 DO
            IF x>15 THEN EXIT FI
            Prtstring()
        OD
    OD
END

PROC Prevrow() ;Move to previous row
    IF y>0 THEN ;if not at first
        y=y-1
        IF y<starty THEN ;screen may move
            starty=y movflg=1
        FI
    FI
    RETURN

PROC Nxtrow() ;Move to next row
    IF y<49 THEN
        y==+1
        IF y-starty>10 THEN
            starty==+1 movflg=1
        FI
    FI
    RETURN

PROC Prevcol() ;Move to previous column
    IF x>0 THEN
        x==+1
        IF x<startx THEN
            startx=x movflg=1
        FI
    FI
    RETURN

PROC Nxtcol() ;Move to next column
    IF x<15 THEN
        x==+1
        IF (x-startx)>3 THEN
            startx==+1 movflg=1
        FI
    FI
    RETURN

PROC Fillbuff() ;Put cell in buffer
    CARD pos
    pos=strs+stringpos()
    MoveBlock(buff,pos,10)
    RETURN

PROC Invbuff() ;Invert all characters

```



# BY STAN OCKERS

```

BYTE i
  FOR i=1 TO 9 DO
    buff(i)=! 128
  OD
RETURN

PROC Shftbuff() ;One space left
BYTE i
FOR i=1 TO Buff(0) DO
  buff(10-i)=buff(buff(0)+1-i)
OD
FOR i=1 TO 9-buff(0) DO
  buff(i)=32 ;Pad with blanks
OD
buff(0)=9 ;Make it full size
RETURN

PROC Addit() ;Insert buff in strs
CARD pos
pos=strs+Stringpos()
MoveBlock(pos,buff,10)
RETURN

PROC Shiftin(BYTE chr) ;Add chr to buff
f
BYTE i
  FOR i=1 TO 8 DO
    buff(i)=buff(i+1)
  OD
  buff(9)=chr ;comes in from right
RETURN

PROC Makedol(BYTE cnt) ;Put in dollars
BYTE i
i=0 DO i==+1
  IF buff(i)=46 THEN EXIT FI
  UNTIL i=cnt+1
  OD
  IF i=cnt+1 THEN
    Shiftin(46) Shiftin(48) Shiftin(4
8)
  ELSEIF i=cnt THEN
    Shiftin(48) Shiftin(48)
  ELSEIF i=cnt-1 THEN
    Shiftin(48)
  FI
RETURN

PROC Entrchr(BYTE chr) ;Put character
BYTE i ;into buffer and display
  Shiftin(chr) Position(col,row)
  Print(buff) cnt==+1
  IF x>maxx THEN maxx=x FI
  IF y>maxy THEN maxy=y FI
RETURN

PROC Coladd() ;Add up a column
REAL sum, addend
BYTE savx,savy
BYTE ARRAY zero
zero="0"
savx=x savy=y
FOR y=0 TO maxy DO
  ValR(zero,sum)
  FOR x=0 TO maxx DO
    ValR(zero,addend)
    Fillbuff()
    IF buff(1)=100 THEN
      StrR(sum,buff) Shftbuff()
      Makedol(buff(0))
      Invbuff() buff(1)=100 Addit()
    ELSEIF buff(1)=222 THEN
      ELSE
        ValR(buff,addend)
        RealAdd(sum,addend,sum)
      FI
    OD
    x=savx y=savy
  RETURN

PROC Rowadd() ;Add up a row
REAL sum, addend
BYTE savx,savy
BYTE ARRAY zero
zero="0"
savx=x savy=y
FOR x=0 TO maxx DO
  ValR(zero,sum)
  FOR y=0 TO maxy DO
    ValR(zero,addend)
    Fillbuff()
    IF buff(1)=222 THEN
      StrR(sum,buff) Shftbuff()
      Makedol(buff(0))
      Invbuff() buff(1)=222 Addit()
    ELSEIF buff(1)=100 THEN
      buff(1)=160 Invbuff()
      ValR(buff,addend)
      RealAdd(sum,addend,sum)
    ELSE
      ValR(buff,addend)
      RealAdd(sum,addend,sum)
    FI
  OD
  x=savx y=savy
RETURN

PROC Fause(CARD cnt) ;Delay for messag
e
CARD i
BYTE t
FOR i=1 to cnt DO t==*t OD
RETURN

PROC Diskerr(BYTE errcode) ;Error trap
Binkline()
IF errcode=170 THEN
  Print("No file of that name")
ELSEIF errcode=138 THEN
  Print("Check disk drive")
ELSEIF errcode=130 THEN
  Print("Specify 'D:' in filename")
ELSEIF errcode=165 THEN
  Print("Bad filename")
ELSE PrintB(errcode)
FI
Pause(10000)
errflg=1 Close(2)
RETURN

PROC Loadsave(BYTE mode) ;Disk I/O
BYTE i
CARD start,pos,temperr
BYTE ARRAY filenm(30)
BYTE ARRAY temp(162)
temperr=Error Error=Diskerr
temp(0)=160 start=temp+1
Binkline()
Print("Filename?") InputS(filenm)
Open(2,filenm,mode,0)
IF errflg=1 THEN errflg=0 RETURN FI
  IF mode=4 THEN
    maxx=InputBD(2) maxy=InputBD(2)
  ELSEIF mode=8 THEN
    PrintBDE(2,maxx) PrintBDE(2,maxy)
  FI
  FOR i=0 TO maxy DO
    pos=strs+(160*i)
    IF mode=4 THEN
      InputSD(2,temp)
      Moveblock(pos,start,160)
    ELSEIF mode=8 THEN
      Moveblock(start,pos,160)
      PrintBDE(2,temp)
    FI
  OD
  Close(2) Error=temperr
RETURN

```

# ROWS CON'T

```

PROC Prterr(BYTE errcode) ;Error trap
Blinkline()
IF errcode=138 THEN
    Print("Check Printer")
ELSE PrintB(errcode)
FI
Pause(10000)
errflg=1 Close(3)
RETURN

PROC Prnt() ;Put it out to printer
BYTE twopage,columns,savex,savey
BYTE ARRAY colms=[0 1 2 3 4 5 6 7
    4 4 5 5 6 6 7 7]
BYTE ARRAY spcing=[0 6 5 4 3 2 1 0
    3 3 2 2 1 1 0 0]
BYTE ARRAY blinks=[1 32 32 32 32 32
    32 32 32]
CARD temperr
temperr=Error
Error=Prterr
savex=x savey=y

... maxx<8 THEN twopage=0
ELSE twopage=1
FI
columns=colms(maxx)
blinks(0)=spcing(maxx)
n(3,"P:",0,0)
... errflg=1 THEN errflg=0 RETURN FI
FOR y=0 TO maxy DO
    FOR x=0 TO columns DO
        Fillbuff() IF buff(1)>128 THEN
            Invbuff() buff(1)=32
        FI
        PrintD(3,buff)
        PrintD(3,blinks)
    OD
    PutD(3,13)
OD
IF twopage=1 THEN
    PutD(3,13) PutD(3,13) PutD(3,13)
    FOR y=0 TO maxy DO
        FOR x=columns+1 TO maxx DO
            Fillbuff() IF buff(1)>128 THEN
                Invbuff() buff(1)=32
            FI
            PrintD(3,buff)
            PrintD(3,blinks)
        OD
        PutD(3,13)
    OD
    /0

PutD(3,13) PutD(3,13) PutD(3,13)
FI
Close(3)
x=savex y=savey
RETURN

PROC Select() ;Menu choices
BYTE chr,casesave
Position(0,0) casesave=Peek(702)
Poke(702,64)
Print("Balc Print Save Load Mode 550")

DO chr=GetD(1)
    IF chr=67 THEN
        Coladd() Rowadd()
        EXIT
    ELSEIF chr=80 THEN
        Prnt() EXIT
    ELSEIF chr=76 THEN
        Loadsave(4) EXIT
    ELSEIF chr=83 THEN
        Loadsave(8) EXIT
    ELSEIF chr=77 THEN
        IF modeflg=0 THEN modeflg=1
        ELSE modeflg=0
        FI EXIT
    FI
UNTIL chr=27
OD
Fillscr() Prtstring() Poke(702,casesave)
RETURN

PROC Main()
BYTE nflg,chr,i,curflg,conkey
CARD pos
Init() Fillscr()
DO
    nflg=0 cnt=1 buff(0)=9 curflg=0
    movflg=0 Fillbuff()
    Position(col,row) Print(buff)
    FOR i=1 TO 9 DO
        buff(i)=32 OD
    DO chr=GetD(1)
        IF chr=27 THEN Select()
        ELSEIF chr=126 THEN
            IF cnt>1 THEN
                FOR i=0 TO 7 DO
                    buff(9-i)=buff(8-i)
                OD
                buff(1)=32 Position(col,row)
                Print(buff) cnt=-1
            FI
            ELSEIF chr=42 THEN
                IF modeflg=0 THEN buff(1)=60
                ELSE buff(1)=94
                FI
                IF x>maxx THEN maxx=x FI
                IF y>maxy THEN maxy=y FI
                Invbuff() Addit() Prtstring()
                EXIT
            ELSEIF chr=28 THEN
                curflg=1 Prevrow() EXIT
            ELSEIF chr=29 THEN
                curflg=1 Nxtrow() EXIT
            ELSEIF chr=30 THEN
                curflg=1 Prevcol() EXIT
            ELSEIF chr=31 THEN
                curflg=1 Nxtcol() EXIT
            ELSEIF chr=60 THEN
                x=0 startx=0 curflg=1
                Fillscr() EXIT
            ELSEIF chr=62 THEN
                x=maxx startx=maxx-3 curflg=1
                Fillscr() EXIT
            ELSEIF chr=94 THEN
                y=0 starty=0 curflg=1
                Fillscr() EXIT
            ELSEIF cnt<10 AND (chr)>47 AND chr<
                58 OR
                chr=45 OR chr=46) THEN
                IF cnt=1 then nflg=1
                FI
                Entrchr(chr)
            ELSEIF cnt<10 AND nflg=0 AND
                (chr)>31 AND chr<123) THEN
                Entrchr(chr)
            FI
            UNTIL chr=155
            OD
            IF cnt>1 AND curflg=0 THEN
                IF nflg=1 THEN Makedel(buff(0)) FI
                Addit() IF modeflg=0 THEN Nxtcol()
                ELSE Nxtrow()
                FI
            ELSEIF curflg=0 THEN
                IF modeflg=0 THEN Nxtcol()
                ELSE Nxtrow()
                FI
            FI
            IF movflg=1 THEN
                Fillscr()
            ELSE
                Prtstring()
            FI
            RETURN
        OD
    RETURN

```

# KANGA BY ALLAN KNAPP

```

1 REM *****
2 REM *      K A N G A      *
3 REM *      by ALLAN KNOPP  *
4 REM *      -----      *
5 REM * PAGE 6 MAGAZINE - ENGLAND *
6 REM *****
7 REM
100 GOSUB 1095
105 GOSUB 1135:GOSUB 965
110 POKE 710,36:POKE 708,38:POKE 709,5
6
115 Z=12:GOTO 120
120 REM P.M. MOVER SUBROUTINE
125 REM SETUP
130 RESTORE 465
135 DIM M0$(6),M1$(6),M2$(6),M3$(6),C$(32)
140 DIM PMMOV$(100),P0$(30),P1$(30),P2$(30),P3$(30),PF$(30),MISMOV$(114):MOV
E=ADR(PMMOV$):FOR K=1 TO 100:READ N
145 PMMOV$(K)=CHR$(N):NEXT K
150 REM NOW READ SHAPE DATA
155 FOR K=1 TO 12:READ N:P0$(K)=CHR$(N):NEXT K
160 FOR K=1 TO 12:READ N:P1$(K)=CHR$(N):NEXT K
165 FOR K=1 TO 12:READ N:P2$(K)=CHR$(N):NEXT K
170 FOR K=1 TO 2:READ N:P3$(K)=CHR$(N):NEXT K
175 FOR K=1 TO 12:READ N:PF$(K)=CHR$(N):NEXT K
180 PMBASE=INT((PEEK(145)+3)/4)*4:POKE 54279,PMBASE:REM SETUP P.M. AREA
185 PMB=PMBASE*256
190 PMR=ADR(P0$)
195 PML=ADR(P1$)
200 PMD=ADR(P2$)
205 PMS=ADR(P3$):REM * PM DATA ADDR
210 PMF=ADR(PF$)
215 POKE 559,46:POKE 53277,3:REM P.M. DMA
220 POKE 704,14:POKE 707,2:POKE 623,1
225 GOSUB 665:TRAP 660
230 REM PROG HERE
235 GOSUB 965
240 POKE 711,56:GOTO 730
245 L=3:SC=1:CC=246:GOSUB 835
250 K=0:TIM=900
255 SOUND 2,240,10,2:SOUND 3,243,10,2
260 POSITION 0,2: ? #6;"BOOM ";SC:POSIT
ION 0,4: ? #6;"SCORE"
265 X=180:Y=94:PMM=PMR:GOSUB 450
270 REM
275 POSITION 5,22:FOR R=1 TO L: ? #6;"":NEXT R:POKE 53278,1
280 POSITION 5,23:FOR R=1 TO L: ? #6;"":NEXT R
285 REM READ STICK
290 S=STICK(0)
295 TIM=TIM-2:POSITION 1,5: ? #6;TIM;"":IF TIM<0 THEN GOTO 545
300 IF PEEK(53279)=3 THEN GOSUB 815:GOTO 730
305 IF S=11 THEN X=X-4:PMM=PML:IF X<40 THEN X=40:GOSUB 450
310 IF S=14 THEN Y=Y-2:X=X+2:PMM=PMD:IF X>192 THEN X=192:GOSUB 450
315 IF S=7 THEN X=X+4:PMM=PMR:IF X>192 THEN X=192:GOSUB 450
320 IF S=13 THEN Y=Y+2:X=X-2:PMM=PMF:IF X<40 THEN X=40:GOSUB 450
325 IF X<40 THEN X=40
330 IF X>200 THEN X=200
335 GX=(X-48)/8:GY=(Y-16)/4:IF GY>20 THEN GY=20
340 IF Y>94 THEN Y=94
345 POKE 711,Z:POKE 77,0
350 LOCATE GX+1,GY+2,C:LOCATE GX,GY+1,D:LOCATE GX,GY+2,E
355 CC=CC+1:IF CC>249 THEN CC=246
360 COLOR CC:PLOT 13,12
365 IF D=250 THEN GOTO 865
370 IF D>33 OR D<48 THEN Y=Y-2:GOSUB 450:Y=Y+2:GOSUB 450
375 IF D>58 AND D<62 THEN Y=Y+4:X=X+2:GOSUB 450
380 IF D>48 THEN Y=Y+2:GOSUB 450
385 IF D<33 AND E<33 THEN GOSUB 515
390 IF D<33 THEN GOSUB 515
395 IF E<33 THEN GOSUB 515
400 IF C<33 THEN GOSUB 515
405 IF D>246 AND D<249 THEN GOSUB 825
410 IF D=90 AND K=1 THEN TIM=TIM+100:GOTO 730
415 IF Y<20 THEN Y=20
420 IF RND(0)>0.99 THEN GOSUB 960
425 Z=Z+16:IF Z>255 THEN Z=12
430 CC=CC+1:IF CC>249 THEN CC=246
435 COLOR CC:PLOT 13,12
440 MSL=MSL+1:IF MSL>10 THEN GOSUB 550
445 GOTO 285
450 A=USR(MOVE,0,PMB,PMM,X,Y,11):A=USR(MOVE,3,PMB,PMS,X+2,Y+9,2)
455 FOR M=30 TO 80 STEP 10:SOUND 1,M,1,0,10:NEXT M:SOUND 1,0,0,0:RETURN
460 REM P.M. MOVE DATA
465 DATA 216,104,104,104,133,213,104,2,4,105,2,133,206,104,133,205,104,133,20,4,104,133,203,104,104,133,208
470 DATA 104,104,133,209,104,104,24,10
1,209,133,207,166,213,240,16,165,201,4,105,128,133,205,165,206,185
475 DATA 0,133,206,202,200,240,160,0,162,0,196,209,144,19,196,207,176,15,132,212,138,168,177,203,164
480 DATA 212,145,205,232,169,0,240,4,169,0,145,205,200,192,128,208,224,166,213,165,208,157,0,208,96
485 REM PLAYER SHAPE DATA
490 DATA 4,6,7,4,12,30,156,124,16,31,0,0
495 DATA 32,96,224,32,48,120,59,62,8,2,48,0,0
500 DATA 12,14,8,13,30,28,28,61,30,36,64,128
505 DATA 62,124
510 DATA 24,56,8,24,28,53,30,28,22,36,72,0
515 POKE 623,4:FOR Y=Y TO 100:SOUND 0,Y,10,10:Y=Y+0.5:GOSUB 450:SOUND 0,0,0,0:NEXT Y
520 L=L-1:IF L<1 THEN GOTO 545
525 POSITION 5,22: ? #6;"":POSITION 5,23: ? #6;"":
530 POSITION 5,22:FOR R=1 TO L: ? #6;"":NEXT R:POKE 53278,1
535 POSITION 5,23:FOR R=1 TO L: ? #6;"":NEXT R
540 X=180:Y=90:POKE 623,1:RETURN
545 TIM=0:GOSUB 815:GOTO 730
550 REM SHOOT MISSILE
555 POKE 53257,0:POKE 53278,1
560 KMP=KMP+1:ON KMP GOSUB 620,625,630,635,640,645,650,655
565 FOR KM=KM TO 225 STEP 10:GOSUB 570:NEXT KM:MSL=0:SOUND 1,0,0,0:RETURN
570 A=USR(MISL,1,PMB,ADR(M1$),KM,YM,6):SOUND 1,KM/8,10,14:IF PEEK(53257)<>1 THEN RETURN
575 SOUND 1,0,0,0:A=USR(MISL,1,PMB,ADR(M1$),225,YM,6)
580 FOR MD=1 TO 5:FOR MM=0 TO 4:SOUND 0,12,50,10:A=USR(MOVE,0,PMB,PMM,180,94,11):ON MM GOSUB 600,605,610,615
585 FOR MD=1 TO 3:FOR MM=0 TO 4:SOUND 0,12,50,10:A=USR(MOVE,0,PMB,PMM,X,Y,11):ON MM GOSUB 600,605,610,615
590 REM KANGA HIT
595 NEXT MM:SOUND 0,0,0,0:NEXT MD:POKE 53257,0:POKE 53278,1:GOSUB 520:GOTO 285
600 PMM=PMR:RETURN
605 PMM=PMR:RETURN
610 PMM=PMR:RETURN
615 PMM=PMF:RETURN
620 KM=56:YM=88:RETURN

```

# KANGA CON'T

```

625 XM=64:YM=80:RETURN
630 XM=72:YM=72:RETURN
  5 XM=80:YM=64:RETURN
640 XM=88:YM=56:RETURN
645 XM=96:YM=48:RETURN
650 XM=104:YM=40:RETURN
655 XM=112:YM=32:XMP=0:RETURN
660 TRAP 40000:POKE 53257,0:POKE 53278
,1:GOTO 285
665 REM MISSILE SETUP
670 RESTORE 695
675 MISL=ADR(MISMOV$):FOR X=1 TO 114:R
EAD M:MISMOV$(X)=CHR$(M):NEXT X
680 REM LOAD MISSILE IMAGE
685 FOR I=1 TO 6:READ M:M1$(I)=CHR$(M)
:NEXT I
690 POKE 705,156:POKE 53260,12
695 DATA 216,104,104,104,133,213,104,1
33,206,104,24,105,128,133,205,165,206,
105,1,133,206,104,133,204,104
700 DATA 133,203,104,104,133,208,104,1
04,133,209,104,104,24,101,209,133,207,
160,0,162,0,134,212,169,252
705 DATA 166,213,240,7,10,10,9,3,202,2
08,249,166,212,49,205,145,205,196,209,
144,30,196,207,176,26
  DATA 132,212,138,168,177,203,164,2
40,5,10,10,136,208,251,164,212,17,
205,145,205,232,169,0,240
715 DATA 0,200,192,128,208,196,166,213
,165,208,157,4,208,96
720 DATA 0,0,3,0,0,0
  RETURN
  GOSUB 965
735 POSITION 0,2:? #6;"score"
740 POSITION 0,3:? #6;STIM
745 IF STIM>HSC THEN HSC=STIM
750 POSITION 0,5:? #6;"HI ":POSITION 0
,6:? #6;HSC
755 POSITION 4,21:? #6;"press start"
760 SOUND 0,240,10,10:SOUND 1,243,10,1
0
765 CC=246
770 CC=CC+1:IF CC>249 THEN CC=246
775 COLOR CC:PLOT 13,12
780 FOR M=1 TO 10:NEXT M
785 Z=Z+16:IF Z>255 THEN Z=12
790 POKE 711,Z
795 IF PEEK(53279)<>6 THEN 770
800 POKE 710,36:POKE 708,38:STIM=0
805 SOUND 0,0,0,0:SOUND 1,0,0,0
810 GOSUB 815:GOSUB 965:POKE 623,1:GOT
O 245
815 FOR M=0 TO 19:COLOR 32:PLOT M,0:DR
AWTO M,23:NEXT M:RETURN
820 FOR M=1 TO 3:SOUND 1,M*10,10,10:NE
XT M:SOUND 1,0,0,0:K=1:RETURN
825 ON SC GOSUB 840,840,845,850,855,84
5
830 FOR MM=1 TO 3:FOR M=1 TO 15:SOUND
1,M*2,10,10:NEXT M:NEXT MM:RETURN
835 POSITION 6,17:? #6;"$+":POSITION
6,18:? #6;")+("):RETURN
840 POSITION 12,8:? #6;"$+":POSITION
12,9:? #6;")+("):RETURN
845 POSITION 11,5:? #6;"$+":POSITION
11,6:? #6;")+("):RETURN
850 POSITION 7,10:? #6;"( &":POSITION
7,11:? #6;"e ?":POSITION 6,17:? #6;"
$+":POSITION 6,18:? #6;")+("
855 POSITION 11,5:? #6;"$+":POSITION
11,6:? #6;")+("):RETURN
860 GOSUB 815:GOSUB 965:GOSUB 910:POSI
TION 6,23:? #6;"Success":GOSUB 935:GOT
O 735
865 SC=SC+1:ON SC GOSUB 875,880,885,89
0,895,900,905:STIM=STIM+TIM:L=L+1:TIM=
TIM+750:GOSUB 815:GOSUB 965
870 FOR S=150 TO 1 STEP -2:SOUND 1,5,1
0,10:NEXT S:GOTO 260
875 REM COLOURS FOR SCREENS
880 POKE 710,54:POKE 708,56:RETURN
885 POKE 710,102:POKE 708,104:RETURN
890 POKE 710,132:POKE 708,134:RETURN
895 POKE 710,164:POKE 708,166:RETURN
900 POKE 710,228:POKE 708,232:RETURN
905 GOTO 860
910 REM MUSIC
915 RESTORE 920:FOR S=1 TO 54:READ T:S
OUND 2,T,10,8:FOR M=1 TO 10:NEXT M:NEX
T S:RETURN
920 DATA 121,121,96,121,121,162,121,12
1,96,121,121,162,121,121,108
925 DATA 96,108,121,128,128,121,108,12
8,162,121,121,96,121,121,162
930 DATA 121,121,96,121,121,162,121,10
8,96,108,108,96,121,121,121,121,12
1,0,0,0,0,0,0
935 SOUND 1,233,14,14:SOUND 2,230,14,1
4:SOUND 3,231,14,10:SOUND 0,228,14,14
940 RESTORE 950:FOR I=1 TO 32:READ C:C
$(I)=CHR$(C):NEXT I:C$(15,15)=CHR$(22)
:X=USR(ADR(C$),10)
945 FOR S=0 TO 3:SOUND S,0,0,0:NEXT S
950 DATA 104,104,104,72,162,57,160,0,1
73,0,210,101,20,141,22,208,141,10,212,
136,208,242,202,208,237,104
955 DATA 56,233,1,208,228,96
960 XMP=RND(0)*6:GOSUB 550:RETURN
965 REM * SCREEN SETUP *
970 POSITION 0,0:? #6;"Kanga" ;+++++
+++++
975 POSITION 0,1:? #6;"
+++++
980 POSITION 0,2:? #6;"
;=+++++
985 POSITION 0,3:? #6;"
<=+++++
990 POSITION 0,4:? #6;"
;)++++++
995 POSITION 0,5:? #6;"
<=H'////e
&+("
1000 POSITION 0,6:? #6;"
;)+("
1005 POSITION 0,7:? #6;"
<=H'e&+
+("
1010 POSITION 0,8:? #6;"
;)+(" ?///
//e"
1015 POSITION 0,9:? #6;"
<=H'e"
1020 POSITION 0,10:? #6;"
;)++++++
+++++
1025 POSITION 0,11:? #6;"
<=H'/////
///// $"
1030 POSITION 0,12:? #6;"
;)+("
1035 POSITION 0,13:? #6;"
<=H'e&+
+("
1040 POSITION 0,14:? #6;"
;)+(" ?'////
/e )+("
1045 POSITION 0,15:? #6;"
<=H'e&+
&+e"
1050 POSITION 0,16:? #6;"
;)+(" )+++++
+++++
1055 POSITION 0,17:? #6;"
<=H'e /////
////e"
1060 POSITION 0,18:? #6;"
;)+("
1065 POSITION 0,19:? #6;"
=H'+++++
+++++
1070 POSITION 0,20:? #6;"
;)+('////////
/// $"
1075 POSITION 0,21:? #6;"
;)+e"
1080 POSITION 0,22:? #6;"
;)+("
1085 POSITION 0,23:? #6;"
?//e"
1090 RETURN
1095 GRAPHICS 2:POKE 711,42:POKE 708,3
6:POKE 710,0:POKE 709,10:POKE 752,1
1100 POSITION 7,3:? #6;"Kanga"
1105 POSITION 9,5:? #6;"BY":POSITION 4
,6:? #6;"ALLAN KNOPP"
1110 ? "PRESS START TO BEGIN
OPTION TO RESTART DURING
GAME"
1115 IF PEEK(53279)<>6 THEN 1115
1120 POSITION 2,1:? #6;"back in a mome
nt"

```

# OBSERVATIONS ON PAPER CLIP

This new word processor has a number of features not found on any other ATARI word processor. It is not all peaches and cream, but PAPER CLIP offers some unique options.

PAPER CLIP'S ability to preview the text before it is printed is unsurpassed. However, there are several other features worthy of note. For example, there are **USER DEFINED** printer codes, specialized "Macro" Files and Koala Graphic capabilities as well as the ability to configure the program for virtually any printer available.

I found a few shortcomings in the program which may influence some prospective buyers. In my estimation, the most serious drawback is that the text buffer will **not** accept key input fast enough to keep up with a very fast typist. I found the key response to be somewhat sluggish. Typing at 50 words a minute will cause the buffer to "bubble over" and drop some characters. For most of us slow typists this will never be a problem. However, to a serious or professional typist this handicap will automatically eliminate PAPER CLIP from contention.

Another disadvantage is that in spite of 155 pages of documentation, I had to read between the lines of the Documentation to specifically adapt the program to my FX-80 Printer. Through experimentation, I have learned a few undocumented or sparsely documented short cuts and/or further enhancements which I will pass along.

It is possible to configure the program to take advantage of the Epson's ability to use proportional type. The procedure is to binary load from DOS the "PRTR.COM" file and then, following the instructions, load in the configuration file for the specified printer. In my case I loaded in the "FX80ALTCNF" file. By going through this file I was able to see what **DEFAULT** values were placed in the file for the various options and printer codes and then record them for future reference and/or possible alteration. There is no record of specific printer code values anywhere in the documentation. The configuration files alone access this information. On the program disk I found that the 4 **USER DEFINED PRINT CODES** were predefined to contain the Graphic Dump Codes for the FX80. I redefined these codes to contain the Proportional and Enlarged ON and OFF codes — and then saved the altered file to another Epson configuration file name of my own.

Further checking the printer configuration file will reveal information about micro spacing and optional pitch fonts which are otherwise not specifically mentioned.

Another discovery was that double column printing will not work correctly unless the formatting line (pitch, margin, etc.) is exactly at the beginning of the text. In other words, set the margins as per instructions but don't put a carriage return at any point between the format line and the beginning of text. If you do, the right column will print first and always be one line above the left column. If you are used to other word processors in which the formatting line stands alone and is terminated by a **RETURN**, this is hard to remember. However if you are careful you will be rewarded with a very fine looking double-column printout.

In writing this review I also found that I could not (while in double column mode) "place" a paragraph symbol at the end of the format line where the text needed to start. I could do this if the text were justified left on the screen — but not at the end of the format line. Since I needed paragraph indentation, the solution was to leave 5 spaces at the end of the format line before beginning to enter text.

The manual suggests that a **CONTROL-SHIFT-DELETE** followed by a **RETURN** will empty the paste buffer. This is not so! Instead, move the cursor to a blank line and follow the **CONTROL-SHIFT-DELETE** with a **SHIFT-DELETE**. This will effectively empty the paste buffer.

Each screen line of PAPER CLIP is ended by a small diamond-shaped mark. This has proved extremely helpful as it serves as a pointer to the number of spaces between the last word on the line and the first word on the next. No longer do I end up with a few gaps here and there nor is it necessary to justify screen output periodically as with the other word processors I have used. This is a fine touch in spite of the fact that it is not documented.

When moving a block of text with PAPER CLIP I thought at first that I had to clear out "room" — or else other text was overwritten. But this is not the case as long as I was not in the over-type mode rather than the insertion mode of text entry. In my early experiments my pastes were sometimes over-pastes.

My original standard for Atari word processors has always been **LETTER PERFECT**. It was sometimes cumbersome to use but has continued to be very flexible. Nevertheless I dropped it last year having grown tired of the incompatibility of LJK and ATARI files.

When **THE WRITER'S TOOL** came out I used it exclusively. It wasn't totally flexible, but it was easy to use. The big drawback to this program was poor file merging capabilities.

PAPER CLIP, in spite of some of my reservations, will be the word processor I will rely on — at least for the Atari 8-bit machine. It isn't **THE Perfect Word Processor** but what it can't do my others can and in my view PAPER CLIP is the best all-around word processor currently available for the ATARI.

— Graham Smith

# COMPUTER QUARTERBACK

**COMPUTER QUARTERBACK** (\$40, SSI, 1981, second edition, by **DAN BUNTEN**) has been around for a number of years, but this was my first introduction to it. Prior to this it was not available on the Atari. I have always been sceptical about games which try to duplicate situations which involve so much human interaction. In a game like football everything depends on each of the individual players' attributes and how they interact as a team in any given situation. To me there are too many variables to actively duplicate in a computer game, certainly a 48k one.

**COMPUTER QUARTERBACK** is a pleasant surprise. The designer utilized his computer space primarily for the development of the play variables. This has limited the amount of computer space for graphics. The graphics display is fairly simple. The primary screen shows a football field to the left with a football in the center. The football is the only graphics on the field, and shows where the current line is. On the edge of the field are two markers denoting first down and current ball position. To the right of this you have a score board which shows score, quarter, down, and time remaining. Above this you find the last play played. It is listed as the offensive play, defensive play, and results. Below the score board is a graphics representation of the two teams' current alignment expressed in traditional "X" and "O" configuration. This is the sum total of the game graphics. The only part which actively moves in a game is the ball marker on the playing field. I was surprised to find that these simple graphics in no way detracted from the game.

The game allows you to play either another player or the computer. The play comes in two versions semipro and pro. The semipro version allows you 18 offensive plays and 14 defensive plays. The game comes with a sheet depicting the course of each play. The pro version allows you 36 offensive plays and 24 defensive plays. Each play is numbered and selected by moving the joystick or paddle until the desired play number appears next to your team.

Part of the excitement in this game is that the offense must press the button three times on a specific play before that choice is activated. This means that both the offensive and defensive player are trying to adjust their lines up to the last minutes. Of course the more times you shift your line the greater the chance that your team will not get the signal and a penalty will be called. Players are allowed to call timeouts and at halftime the teams' statistics are shown. As this is a real time game players may choose the length of time of each quarter, the computer is set to play best with 10 minute quarters. In this game the players have also been given the option to create their own teams or send in for discs from SSI, which have stats for the major teams in 1980. When using the draft, player quality will effect all aspects of the game. The pro version also allows you to create a game film, if you have a 80 column printer. This game film lists play by play down situations, yards to first down before play, absolute yard mark, play of both teams and headings for quarter and score. Basically I find this game a very fast paced game which leaves me with the real flavor of football, all you need is the roar of the crowd.

— Nick Chrones

# A Programmer Test Drives a 130XE

Over four years ago I bought my first computer, an Atari 800 which came with a whopping 16k of RAM. It served me well, but this past winter it was starting to let me down. Imagine my joy when I heard that the highly touted new Atari 130XE computers were going to be available almost immediately at a price one sixth of my original computer. Better yet, they would have 128k of RAM and be completely compatible with existing Atari 8 bit machines. What choice did I have but to rush out and buy one?

The first thing a former 800 owner notices when he brings home his 130XE is its size, or lack of it. It is tiny and feather light compared to the old 800! After using it, one discovers that it handles very much like an 800XL. This, of course, means one has to put up with booting up a translator for some software, and I have even come across some older programs that absolutely refuse to work no matter what.

However, I think the new programs which will run only on this new machine will make those old ones pale in comparison, so this incompatibility will only be short lived. Besides, the idea of a modifiable operating system isn't all bad. For example, the Eugene, OR Translator includes an option for fast floating point routines, and with some of the mathematics programs I use, the difference is really noticeable. In fact, I suggest some talented machine language programmers to look into rewriting the printer driver portion of the OS so it can use a bank or two of the extra 130XE memory as a printer buffer. Then an application program could print to the printer as fast as memory transfer can take place, and while the user is doing something else, the printer driver can send the data out as the printer becomes ready for it, perhaps during VBI or by some other interrupt mechanism. This type of multitasking could be a very saleable project, so how 'bout it you guys with the know how?

The manual for the 130XE is quite good. It gives a description of all the BASIC commands, and includes several example programs much as the old BASIC reference manual did (only this manual touches on Player Missile graphics). The manual includes all kinds of technical information and pinout diagrams of the 130XE. It also explains how to access the extra memory banks.

Perhaps the most noticeable physical feature for this programmer in changing to the 130XE was the different keyboard. It has a 'spongy' feel, but it grows on you. At first I didn't like it, but now I really do. Moving to it from an 800, I especially like the larger shift keys, and the moving of the inverse video key to the corner. A nice touch is having the graphics characters on the edge of the keys ala the old Commodore machines — this feature will likely save me a lot of time while programming using character graphics.

The extra memory in the 130XE is the most useful and noticeable functional difference, and will no doubt become even more useful in the future. Already the RAMDISK of DOS 2.5 makes excellent use of it, but how about somebody coming out with a RAMDISK which will work with double density (are you listening OSS??) The RAMDISK also makes a disk based language like C much quicker and nicer to use. I have already seen an ad for BASIC XE which claims to use the extra memory to give the programmer a sizeable 60,000 bytes of programming space (I know of a BBS programmer who should be happy to hear that!). I foresee no end to the data bases, spread sheets, and word processors which all squeeze something out of the extra RAM. I wrote a sector copier on the 130XE to copy a whole single density disk on a single pass, allowing easy multiple copies to be made. I think a few club librarians will be eager to try that out (it also does a double density disk in two passes). The extra memory is very easy to use from the programmer's standpoint.

Another really nice feature an old 800 owner will notice is that the 130XE, like the XLs has a true hardware reset key, which means you can recover from almost any crash without loss of program and temper. I have managed to lock the machine up several times when programming in ACTION!, and each time I was able to recover my program text. Additionally, the RAMDISK is also preserved through a reset. The net result of all this is that in many cases where with the old 800 I would have lost my work, with the 130XE I was safe.

I have a couple of gripes, though. First of all, I'm still not too hot about the idea of the cartridge slot being hidden behind the machine. It would have been handier for it to be on the side. Also, the joystick ports are recessed deep into the machine, making my speech synthesizer not want to fit without heavy removal of plastic on the connectors. Finally, wouldn't it have been nicer if the BASIC would not be active UNLESS the Option button was held down? It seems I am always booting programs which squawk if the BASIC is present. Actually, it is an easy matter for programs to try to remove the BASIC first by altering the memory bank selection byte, and then they should squawk if a cartridge still appears to be present. For example, I did this in a sector copier for the 130XE that I wrote.

Who will want to buy a 130XE? I think all those out there who have mastered their 800s and want a little more power will want to get one. When more programs come out which require the extra memory, those with XLs may also wish to consider upgrading. After all, the cost is not prohibitive. What about the argument that anyone buying a computer should get a 16 bit machine, specifically an ST? First of all, it is quite a bit more investment, and all your old software won't work. It further remains to be seen how the STs will catch on and what software will come out. So, unless you have lots of money and are prepared to start again from scratch, if you want a new computer you better stick with a 130XE.

To sum up, although there are a few minor ouches, I have lots of praise for the 130XE. I heartily recommend that anyone with an old, ailing 800 buy one. You won't be sorry. Incidentally, anyone out there interested in an old, faithful Atari 800 . . . ?

— Dale Lutz

## ROWS AND COLUMNS

**Rows and Columns** is a program to aid in producing financial reports in row and column format. It consists of an array of cells 16 columns wide by 50 rows high. Each cell is 9 characters wide and can contain one of four types of data:

- 1) Alphanumeric data - Used mainly to label rows or columns, it starts with a non-numeric character and its numeric value is zero.
- 2) A number - Once you start to enter a number, you can enter nothing else. The number will be converted to dollars and cents format.
- 3) A column sum - Sums show up in inverse characters on the screen along with an arrow (pointing up for columns). Sums are cumulative. If more than one appears, each contains the sum of all numbers above it (not including previous sums)
- 4) A row sum - Row sums contain all number cells to the left of the sum cell.

### Mode of Entry

There are two modes of entry; by rows and by columns. You can toggle between them by pressing ESC and then 'M' (for Mode). The mode of entry is indicated at the top of the screen. After a cell is filled the cursor will be positioned down one for column entry or to the right one cell for row entry. Pressing '\*' will indicate you want a sum to occupy the present cell. If you are in row entry mode it will be a row sum, a column sum if in Column entry mode.

**Moving Around:** You can move around the array of cells by using the arrow keys (CTRL arrow). The cell for entry is to the left of the cursor. The 'caret' key will move you to the top of the page; the 'lesser than' key all the way to the left side. The 'greater than' key positions you at the rightmost column containing an entry.

**Calculation:** Sums are updated only when you specify to do so. To update the sums press ESC followed by 'C' (Calculate). Rows are summed first and then columns, leading to these two rules:

- 1) No column sums are included in a row sum, (column sums are done second).
- 2) Row sums are treated as normal numbers when doing column sums.

Since one space of the sums is taken by the arrow, the highest value that can be summed is 99999.99. If you're doing tabulations where larger sums will appear you can probably afford to by a better program.

**Saving/Loading Data:** Saving or loading is done by pressing ESC followed by a 'L' or 'C'. You will be prompted for a filename. Use the full filename (D:CASH.CAL).

**Printing:** Use ESC followed by 'P' to print to a printer. 80 columns is assumed and if required, the printing will be split in two sets of columns separated by three blank lines.

— Stan Ockers

# SKYPLOT

PLAN **ATARI** UM - ACE-ironomy Disk

The best things in life are free, right? You doubt it? Look up, at the evening or morning sky. You can share Newton's and Galileo's wonder at the magnificent sky under which we live our lives, and enjoy scenes which have produced awe and delight in a thousand generations of humanity.

But "You can't tell the players without a program". Now we have a program for you in the ACE Library. It's called Plan **atari** um. It will show you the sun, moon and planets as they can be seen from any place on earth this side of the Poles, for any time you desire, ranging from this evening to 2001, from your birth day (and time) to that of Caesar, Buddha or the U.S.A.

See the moon and planets in the skies of Asia as Marco Polo saw them on the silk caravan route. Watch a solar eclipse as astronomers will see it from Mauna Kea, Hawaii in 1991. Or follow the sun around the sky until it sets . . . in the North, then rises again an hour or two later, still in the North!

Boot up with BASIC, and run Plan **atari** um. Follow the simple directions for adjusting your color screen, then look at the sample half-sky display of planets . . . at noon!

Only on your Atari can you see the planets lined up in the full light of day, and watch the solar system flow, hour to hour, or year to year. Set the time to noon of any day, then let the program cycle a month at a time. Watch Venus as it moves around the sun in its orbit, while Mercury goes twice as fast in its much smaller path around our star. To watch Mercury circle the sun, set the cycle step to 10 days by pressing the "clear" (don't Shift) key. You will see the sun fixed in the (noon) sky while its two nearest satellites move around it in steps — and you watch, moving with its third satellite.

Each planet is in the sky overhead once each day, but the time of day varies for each with its place in its own orbit. Saturn takes more than 100 times longer to complete one orbit than does Mercury. So Mercury, like Venus, moves rapidly across the sky night to night, then disappears into the glare of daylight. Saturn seems almost fixed in place like a star. It's in the night sky for almost six months. Then in the daylight half of our sky it's fixed for the next six months, almost like Orion or the Pleiades.

It's interesting to explore the way the planets look to people in the southern hemisphere. You can 'go' to Tasmania (42 S, 148 E) or the Straits of Magellan (53 S, 71 W) to watch the sun, moon and planets moving across the northern sky. Then you could cycle the display in latitude (hit + then L, then "Insert" (no Shift!) once or twice), after picking a dawn sky full of planets in the east. Try this in late June and December, then in either March or September.

This will demonstrate the change of seasons as the sun moves from the winter to summer solstices, rising over the tropic of Cancer, then over the Equator, finally — six months later — over the tropic of Capricorn. Notice how short the day is in June when you are in Christchurch, New Zealand, and how long Christmas Day is to folks who live there — it's their Summer Holiday time!

One last suggestion: try 'going' to the Galapagos Islands (0 deg. lat., 91 W) on, say, November 21st of 1984, then set the time to dawn and display the eastern horizon. Start a monthly cycle (+ , then M) and let it run for a year or so. Back up, if you like, using the - key. Try using a 10-day interval ("Clear" key — no Shift!). Try the same thing, but from Hawaii (19 N, 155 W) or Tahiti (18 S, 149 W). Then try it from Iceland or the Yukon, and your own back yard. Enjoy!

— Dick Barkley  
President

# KANGA

Kanga, by Allan Knopp comes to us from the English Atari magazine, **PAGE 6**. "Kanga is a maze game in which you control a jumping kangaroo moving around various platforms. In order to progress to the next screen you must first get to the sparkling jewel in the center of the screen and jump up to touch it. This causes a further section of the platform to appear which will enable you to reach the exit.

"Coming too close to the edge of the platform will cause the kangaroo to fall into space and lose a life. Lives can also be lost by being hit by a laser which fires at random from the windows in the left hand wall. There is a time limit which causes the game to end when it reaches zero. The score remaining at the end of each screen is added to the total and displayed at the end of the game as the final score.

# ATARI 520 ST

Nine users here in Eugene have had their 520 STs since mid-July. I've been "using" mine even more than my XE. I say "using" since there really isn't much software for it yet. It came to us with only the GEM Desktop, the operating system, and Atari LOGO programming language. I've still not begun to learn LOGO, but I intend to dabble in it a bit. Since then we've received about a disk full of public domain demos and advertisements sent to us from the San Leandro Computer Club (their July and August disks). These demos really whet my appetite for the powerful programs to come!

There is a terminal emulator built into the ST operating system. I've been using this and a 1200 baud modem to talk to the BBS's around town. This has been a lot of fun. I've also been making a lot of copies of the public domain stuff for the other ST users in town. Boy is disk management easy and QUICK! Even with one disk drive, although I usually remind the user to bring their drive to speed copying.

The keyboard is a great improvement on the IBM PC keyboard I've used at work for the past 3 years. It does lack one thing my old 800 had: A decent space on which to rest my palms while typing. Looking at my XE, I see it also has nearly zero space between the spacebar and the edge of the machine. But I didn't notice this lack on the XE since the space bar is not wide and there is an area on each side of the space bar on which to rest my palms when I'm not acutally striking keys.

A couple of other minor things: It took me a little while to figure out how to set the clock and calender on the ST. The manual did not make clear to me I had to "backspace" over the default settings before typing in the new. And when I try to write to an un-formatted disk, the OS tells me the drive is not responding. Seems like the OS could detect an unformatted disk. It made me think something was wrong with my drive. I kept powering down and trying to get the drive to "respond"!

Our experience with DOA STs seems to match that of other clubs. About 30% of them did not work upon first boot up. But it appears there is a fix which has so far been 100% successful. In shipping (or something?) the ROM chips become loose. All you need to do (if you have this problem) is get in there and re-seat all the socketed ICs removing the adhesive pads from the two 68-pin square chips. Here's the procedure (reported by Joe Wrobel in the August, 1985 issue of **The Acorn Kernel**) which, we are told is approved by Atari and assured it will not void our warranty:

"You start by removing the 6 screws accessible through the bottom of the unit. This frees the top cover which can be set aside. Removing the cover reveals the keyboard which is sitting atop a metal shield protecting the main PC board underneath. The keyboard is connected to the main board by a keyed 18-pin in-line connector. To remove the keyboard, one just disconnects this connector and undoes the tape which holds the cable to the metal shield. The keyboard can now be set aside. The top metal shield is held down to the main board by three screws and a series of twisted taps which come up from the metal shield beneath the main board and fit into slots in the top shield. Two of these tab-slot pairs (one on each side) are soldered together. After removing the screws and untwisting the other tabs, these bonds are desoldered to remove the top metal shield.

"The top shield has copper tape covering holes in its top and rear. These holes are there for access to and output from the RF modulator which was originally planned into the machines and has subsequently been removed (at least for US purchasers) . . .

"I reseated the DIP ICs by first using a small screwdriver to gently lift each end of the IC partly out of its socket, and then firmly pressing them back into place. The two square ICs were removed by prying them out with a small tool like a jeweler's screwdriver inserted into a corner of the IC socket. I peeled the double-sided adhesive off the chips and removed any remnants with my finger nail, then snapped them back in their respective sockets. If you should have to go through this procedure with your machine, be very careful to note the orientation of the square chips in their sockets before you remove them, and work on them one at a time so they can't get switched.

"Re-assembly of the ST goes rather quickly. When replacing the external screws, note the three longer ones are used to fasten the case at the rear of the machine . . .

A user here in Eugene called Atari for a warranty replacement rather than to open up his machine. Atari was very helpful and air-shipped him a replacement upon his phone call (and before they had even received the old machine!). His replacement arrived the day after he mailed off the old one.

— Jim Bumpas

### Atari Computer Enthusiasts

A.C.E. is an independent, non-profit and tax exempt computer club and user's group with no connection to Atari Corp. We are a group interested in educating our members in the use of the Atari Computer and in giving the latest News, Reviews and Rumors.

**All our articles, reviews and programs come from you, our members.**

Our membership is world-wide; membership fees include the A.C.E. Newsletter. Dues are \$14 a year for U.S., and \$24 a year Overseas Airmail and include about 10 issues a year of the ACE Newsletter.

**Subscription Dep't:** 3662 Vine Maple Dr., Eugene, OR 97405.

**\*\*President—** Dick Barkley, 2907 Wingate, Eugene, OR 97405  
503-344-5843

**Vice Pres—** Larry Gold, 1927 McLean Blvd., Eugene, OR 97405  
503-686-1490

**Librarian—** Ron and Aaron Ness, 374 Blackfoot, Eugene, OR 97404  
(503) 689-7106.

**Editors —** Mike Dunn, 3662 Vine Maple Dr., Eugene, OR 97405

503-344-6193

—Jim Bumpas, 4405 Dillard Rd., Eugene, OR 97405

503-484-4746

**E.R.A.C.E. (Education SIG Editor)—** Nora Young, 105 Hansen Lane,  
Eugene, OR 97404 / 503-688-1458

Send 50c stamps or coin (\$1 overseas) to the Ness' for the new,  
updated ACE Library List — new in May 84 !

### Bulletin Board (503) 343-4352

On line 24 hours a day, except for servicing and updating. Consists of an 800 XL, 2 double-density double sided disk drives and 2 double-sided, double-density, 80-track disk drives, an Epson MX80 printer, a 1200 baud Prentice P212ST modem, running the Mindlink Bulletin Board software distributed by SofMark.

### Best of ACE books

Volume 1 contains bound issues of the ACE Newsletter from the first issue, Oct 81 to June of 1982

Volume 2 covers July 1982 to June 1983

Only \$12 each (\$2 extra for Airmail). Available only from:

George Suetsugu

45-602 Apuapu St

Kanoehe, HI 96744

## TYPESETTING FROM YOUR COMPUTER

**ATARI OWNERS:** If you have a modem, text editor, and communications program to send ASCII files, you should consider the improved readability and cost savings provided by **TYPESETTING** your program documentation, manuscript, newsletter, or other lengthy text instead of just reproducing it from line printer or daisy-wheel output. Computer typesetting by telephone offers you high quality, space-saving copy that creates the professional image you want! Hundreds of type styles to choose from with 8 styles and 12 sizes "on line." And it's easy to encode your copy with the few typesetting commands you need.

### COMPLETE CONFIDENTIALITY GUARANTEED

— Bonded for your protection —

### PUBLICATION DESIGN, EDITING, & PRODUCTION

## Editing & Design Services

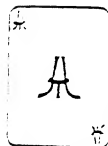
**30 East 13th Avenue Eugene, Oregon 97401**  
**Phone 503/683-2657**

### SortFinder Index

A composite index of Atari related articles in four popular computer periodicals, including ACE. Volume 1 covers April, 1981 to June, 1983. Volume 2 covers July, 1983 to December, 1985. Only \$6 per printed copy or \$11 per disk copy for ACE members:

Jim Carr 2660 S.W. DeArmond

Corvallis, OR 97333



ATARI  
COMPUTER  
ENTHUSIASTS

3662 Vine Maple Dr. Eugene OR 97405

# FIRST CLASS MAIL

